

# Informe Ejecutivo - Migración a MuleSoft API Gateway

**Fecha:** Viernes, 28 de Noviembre de 2025  
**Versión del Proyecto:** 0.2.0-SNAPSHOT  
**Estado Actual:** MVP Operativo con Providers Directos  
**Estrategia:** Migración Zero-Downtime con Canary Deployment

## Resumen Ejecutivo

El **Signature Router** actualmente utiliza **conectores directos** a providers externos (Twilio, Firebase, Biometric APIs). Esta arquitectura es **temporal** y será reemplazada por **MuleSoft API Gateway** como única capa de integración.

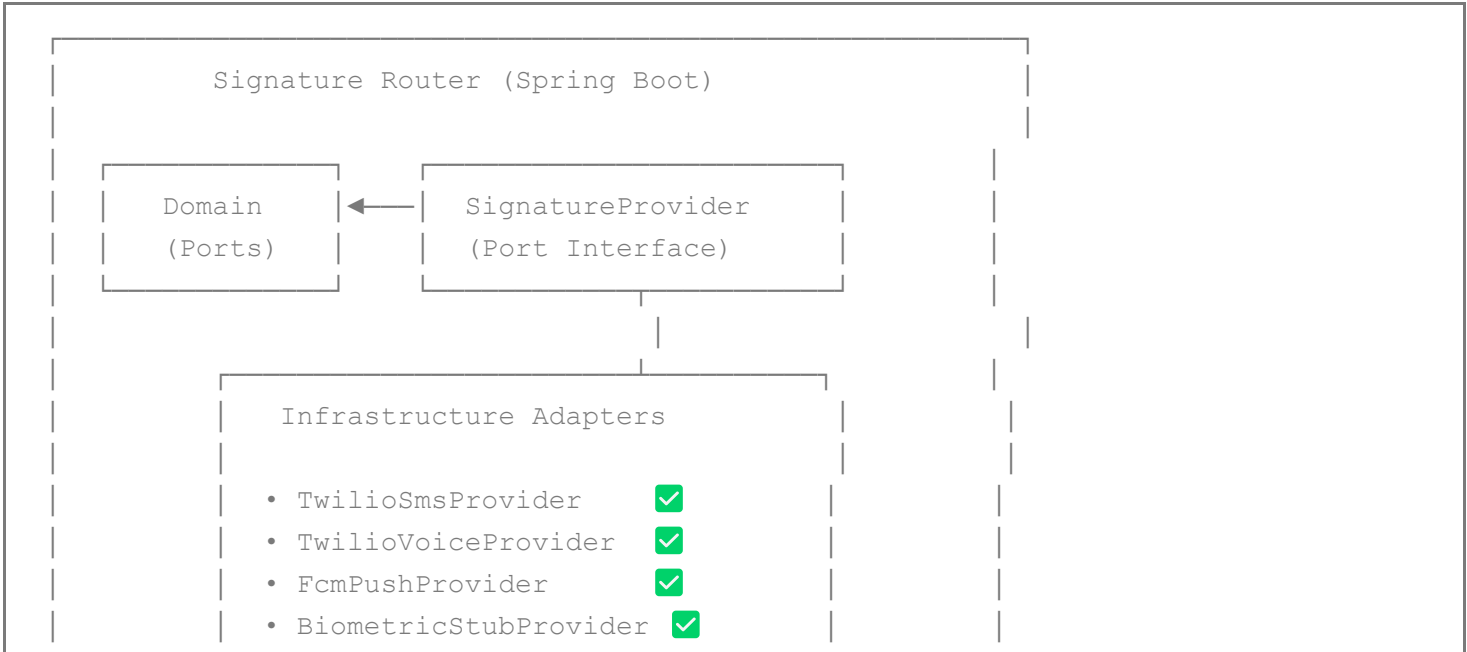
### Ventaja Clave

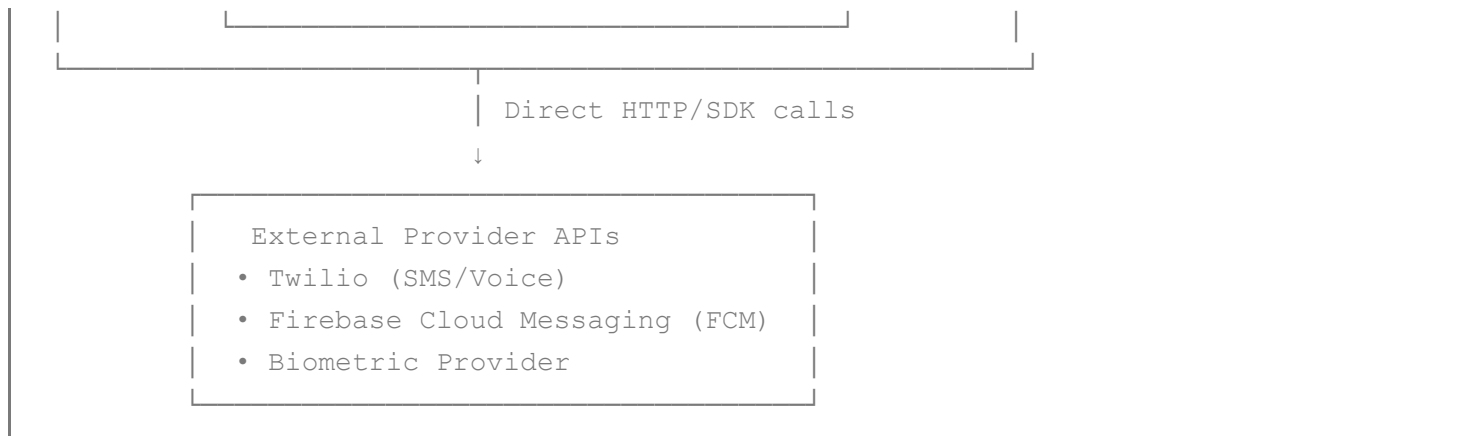
La **Arquitectura Hexagonal** implementada permite migración **Plug & Play** con:

- ✔ Zero impacto en lógica de negocio
- ✔ Zero downtime (canary deployment)
- ✔ Rollback inmediato si hay problemas
- ✔ Reutilización total de resilience patterns (Circuit Breaker, Retry, Metrics)

## Arquitectura: Estado Actual vs. Target

### Estado Actual (MVP - Implementado)

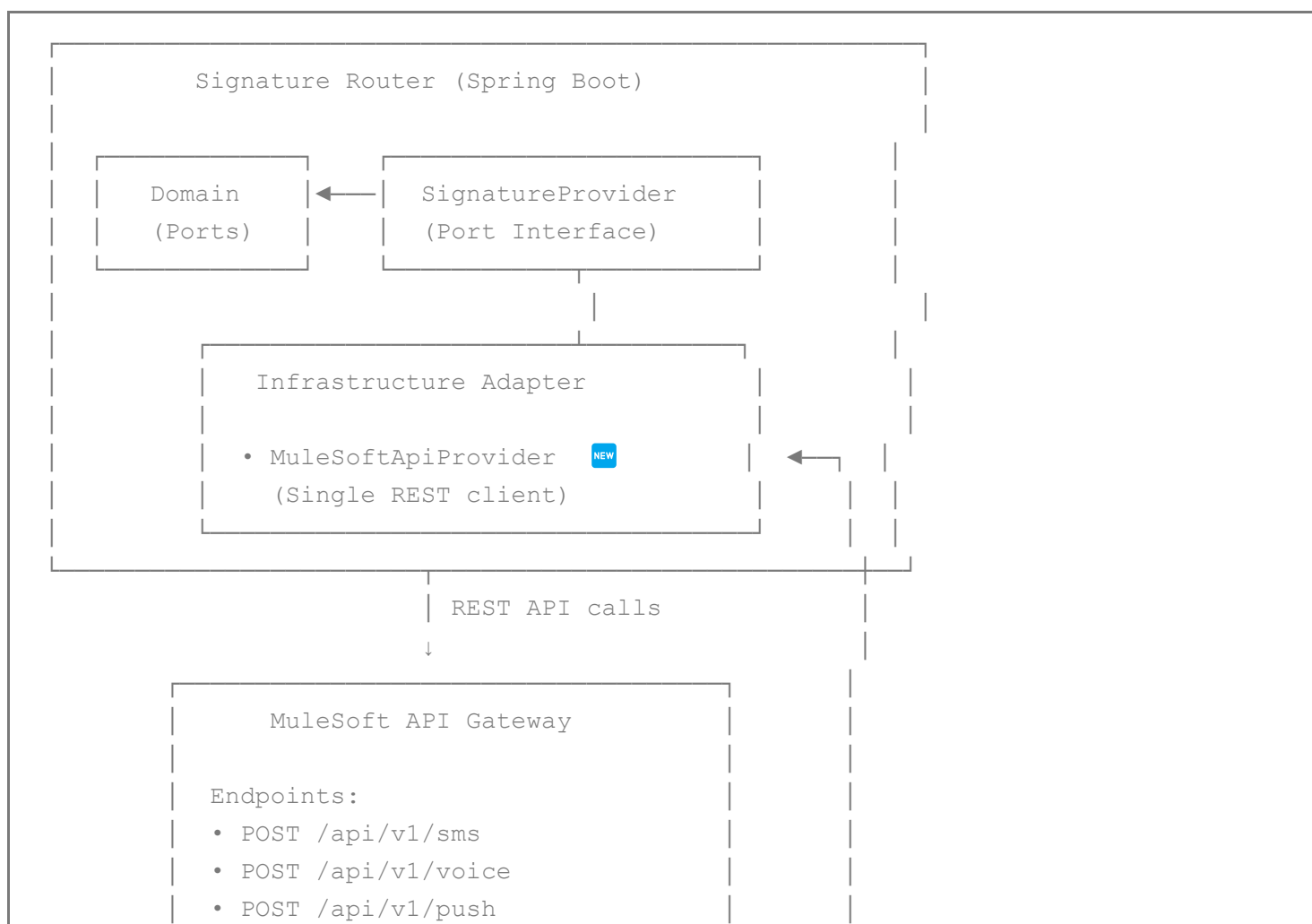


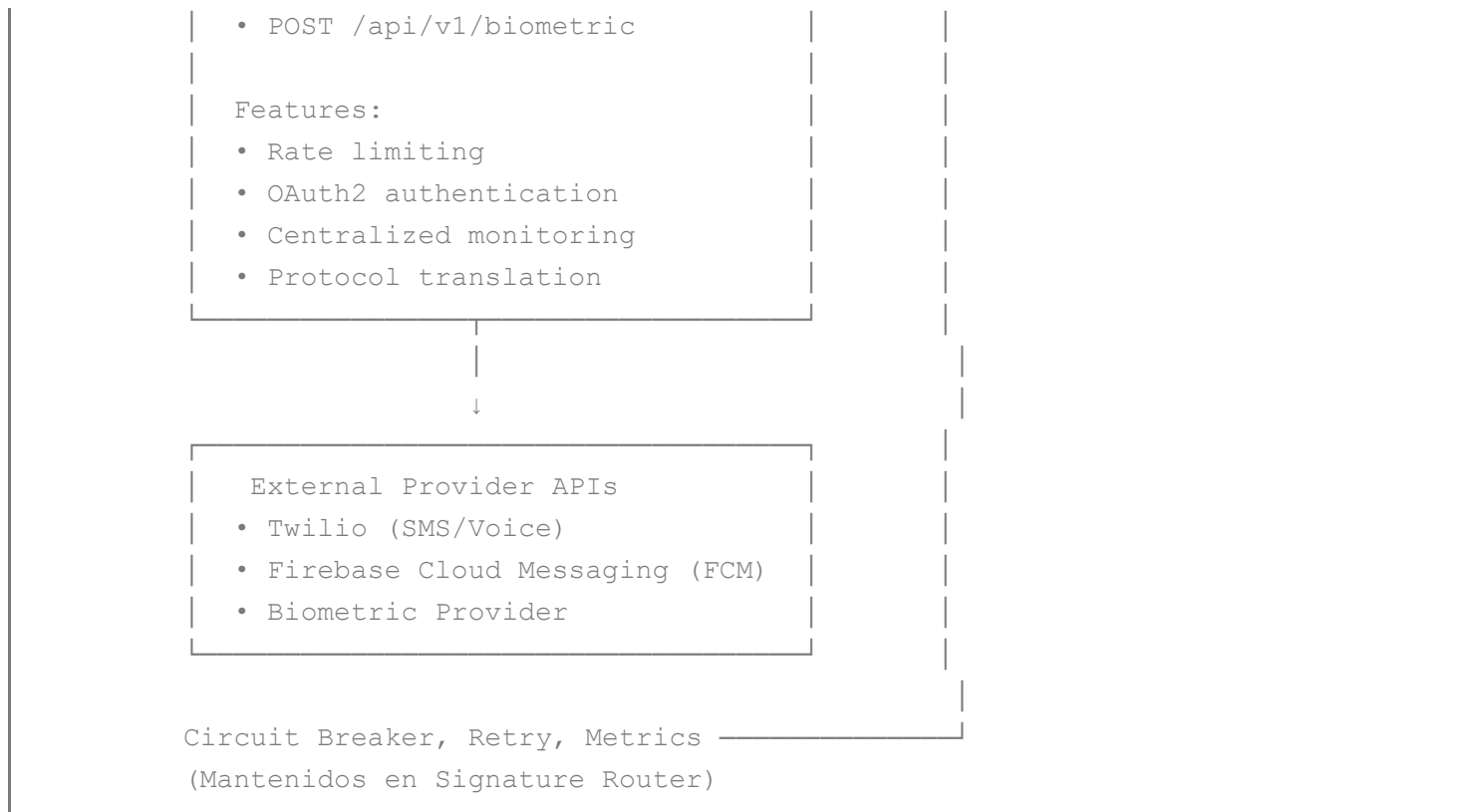


## Características:

- Conectores directos (4 adapters)
- Resilience4j (Circuit Breaker, Retry)
- Métricas Prometheus por provider
- Acoplamiento a SDKs externos (Twilio SDK, Firebase Admin SDK)
- Múltiples puntos de integración

## Estado Target (Post-Migración MuleSoft)





### Características:

- ☒ **Un solo adapter** (`MuleSoftApiProvider`)
- ☒ **Zero acoplamiento a SDKs** externos
- ☒ **Single integration point** (MuleSoft)
- ☒ **Resilience patterns conservados** en Signature Router
- ☒ **Métricas unificadas** (mismo Prometheus)
- ☒ **API Gateway centralizada** (rate limiting, auth, monitoring en MuleSoft)

### Plan de Migración (4 Fases)

#### ☒ Fase 0: MVP con Providers Directos (COMPLETADA)

Status: ☒ DONE

Duración: 5 sprints (10 semanas)

Fecha: 2025-11-01 → 2025-11-28

#### Deliverables Completados:

- ☒ Arquitectura Hexagonal con `SignatureProvider` port
- ☒ 4 adapters implementados:
  - `TwilioSmsProvider` (SMS)
  - `TwilioVoiceProvider` (Voice)

- `FcmPushProvider` (Push)
- `BiometricStubProvider` (Biometric)
- ☒ Resilience4j Circuit Breaker per provider
- ☒ Retry logic con exponential backoff
- ☒ Timeout configuration
- ☒ Prometheus metrics tracking
- ☒ Provider health checks
- ☒ Degraded mode manager

### Propósito:

MVP funcional mientras MuleSoft API specs están en desarrollo.

---

### Fase 1: Diseño de Contrato MuleSoft API

**Status:**  **PENDING** (esperando especificaciones MuleSoft)

**Duración Estimada:** 2-3 semanas

**Esfuerzo:** 40-60 horas

**Responsable:** MuleSoft Team + Signature Router Team

### Prerequisites

- ☐ **OpenAPI 3.0 Specification** definida por MuleSoft team
- ☐ **Authentication mechanism** (OAuth2, API Key, mTLS)
- ☐ **Rate limiting policies** definidas
- ☐ **Error response format** estandarizado
- ☐ **Timeout SLAs** por endpoint (ej: P99 < 500ms)
- ☐ **Monitoring & observability** requirements

### Deliverables

#### 1. OpenAPI 3.0 Specification completa

- Endpoints: `/api/v1/sms`, `/api/v1/voice`, `/api/v1/push`, `/api/v1/biometric`
- Request/Response schemas
- Error codes estandarizados
- Security schemes

#### 2. API Contract Tests (Pact o Spring Cloud Contract)

- Consumer contracts definidos por Signature Router
- Provider contracts validados por MuleSoft team

- CI/CD integration

### 3. Authentication Credentials (dev/uat/prod)

- OAuth2 client credentials
- API keys
- Vault secrets configuration

### 4. SLA Agreements

- Response times (P50, P95, P99)
- Availability targets (99.9%)
- Rate limits (requests/second por app)

## Ejemplo de Contrato MuleSoft (Draft)

```
# OpenAPI 3.0 - MuleSoft SMS Endpoint
POST /api/v1/sms:
  summary: Send SMS challenge
  security:
    - OAuth2: [signature.send]
  requestBody:
    content:
      application/json:
        schema:
          type: object
          required: [recipient, message]
          properties:
            recipient:
              type: object
              required: [phoneNumber]
              properties:
                phoneNumber: { type: string, pattern: '^\\+[1-9]\\d{1,14}$' }
                countryCode: { type: string }
            message:
              type: object
              required: [body]
              properties:
                body: { type: string, maxLength: 160 }
                from: { type: string }
          metadata:
            type: object
            properties:
              transactionId: { type: string, format: uuid }
              timestamp: { type: string, format: date-time }
  responses:
    '200':
      description: SMS sent successfully
```

```

    content:
      application/json:
        schema:
          type: object
          properties:
            messageId: { type: string }
            status: { type: string, enum: [SENT, QUEUED] }
            timestamp: { type: string, format: date-time }
'400':
  description: Invalid request
'429':
  description: Rate limit exceeded
  headers:
    Retry-After: { schema: { type: integer } }
'503':
  description: Provider unavailable

```

## Fase 2: Implementación MuleSoftApiProvider

Status:  NOT STARTED

Duración Estimada: 2 semanas (1 sprint)

Esfuerzo: 80-100 horas

Story Points: 8 SP

### Tasks Detalladas

#### 1. Crear MuleSoftApiProvider Adapter (3 SP - 30-40 horas)

```

@Service
@RequiredArgsConstructor
@ConditionalOnProperty(name = "providers.mulesoft.enabled", havingValue = "true")
public class MuleSoftApiProvider implements SignatureProvider {

    private final WebClient muleSoftWebClient;
    private final MuleSoftConfig config;
    private final ProviderMetrics providerMetrics;

    @Override
    @CircuitBreaker(name = "muleSoftProvider")
    @Retry(name = "muleSoftRetry")
    @TimeLimiter(name = "muleSoftTimeout")
    public ProviderResult sendChallenge(
        SignatureChallenge challenge,
        ChallengeRecipient recipient
    ) {
        // Implementación
    }

```

```
}
```

## Archivos a crear:

- MuleSoftApiProvider.java (adapter principal)
- MuleSoftConfig.java (configuration properties)
- MuleSoftChallengeRequest.java (DTO request)
- MuleSoftChallengeResponse.java (DTO response)
- MuleSoftErrorHandler.java (error mapping)

## 2. Configuration Management (2 SP - 20-25 horas)

```
# application.yml
providers:
  mulesoft:
    enabled: false # Feature flag
    base-url: ${MULESOFT_BASE_URL}
    api-token: ${MULESOFT_API_TOKEN} # From Vault
    endpoints:
      sms: /api/v1/sms
      voice: /api/v1/voice
      push: /api/v1/push
      biometric: /api/v1/biometric
    timeout-seconds: 5
    retry-max-attempts: 3

resilience4j:
  circuitbreaker:
    instances:
      muleSoftProvider:
        failure-rate-threshold: 50
        wait-duration-in-open-state: 30s
        sliding-window-size: 100
```

## Archivos a modificar:

- application.yml
- application-local.yml
- application-uat.yml
- application-prod.yml
- Vault secrets configuration

## 3. Resilience Integration (2 SP - 20-25 horas)

- Circuit Breaker configuration
- Retry policy (align con MuleSoft rate limits)
- Timeout configuration (basado en SLAs)
- Fallback behavior (degraded mode)

#### 4. Testing (1 SP - 10-15 horas)

- **Unit tests:** Mock MuleSoft API (15+ tests)
- **Integration tests:** WireMock/Testcontainers
- **Contract tests:** Pact consumer tests
- **Performance tests:** Load testing (100 req/s)

#### Archivos de tests:

- `MuleSoftApiProviderTest.java` (unit)
- `MuleSoftApiProviderIntegrationTest.java` (integration)
- `MuleSoftApiProviderContractTest.java` (contract)

---

### Fase 3: Deployment Paralelo (Canary)

Status:  NOT STARTED

Duración Estimada: 2 semanas

Esfuerzo: 60-80 horas

Responsable: DevOps + Signature Router Team

Estrategia: Blue-Green con Canary Deployment

Semana 1: Despliegue inicial (10% tráfico)

```
# Configuration: 10% canary
providers:
  # Legacy providers (mantener activos)
  twilio:
    enabled: true
  push:
    enabled: true
  voice:
    enabled: true

# New MuleSoft provider
mulesoft:
  enabled: true
  canary-percentage: 10 # Solo 10% de tráfico
```

Tasks:

- 1. Deploy MuleSoftApiProvider a producción (feature flag OFF)
- 2. Enable feature flag para 10% de tráfico
- 3. Monitor métricas 24/7:
  - Latency (P50, P95, P99)
  - Error rate
  - Success rate
  - Circuit breaker state
- 4. Comparar MuleSoft vs. direct providers (A/B testing)

Semana 2: Incremento gradual

Día	Canary %	Monitoreo	Rollback Plan
Lunes	10%	Baseline establecido	Automático si error rate > 5%
Miércoles	25%	48h estable con 10%	Feature flag OFF
Viernes	50%	48h estable con 25%	Feature flag OFF
Siguiente semana	75%	48h estable con 50%	Feature flag OFF
Final	100%	1 semana estable con 75%	Rollback a providers directos

Métricas de Monitoreo

Grafana Dashboard: MuleSoft vs. Direct Providers

```
# Latency comparison
histogram_quantile(0.99,
  sum(rate(signature_provider_duration_seconds_bucket{provider="MULESOFT"}[5m]))
by (le)
) vs
histogram_quantile(0.99,
  sum(rate(signature_provider_duration_seconds_bucket{provider="TWILIO"}[5m])) by
(le)
)
```

```
# Error rate comparison
sum(rate(signature_provider_errors_total{provider="MULESOFT"}[5m])) vs
sum(rate(signature_provider_errors_total{provider="TWILIO"}[5m]))

# Success rate comparison
sum(rate(signature_provider_success_total{provider="MULESOFT"}[5m])) vs
sum(rate(signature_provider_success_total{provider="TWILIO"}[5m]))
```

## Criterios de Éxito para Avanzar

- ☒ Latency regression < 10% (MuleSoft vs. directo)
- ☒ Error rate parity (MuleSoft error rate ≤ directo)
- ☒ Zero customer complaints
- ☒ Circuit breaker estable (no opens inesperados)
- ☒ 48 horas sin incidentes

## Rollback Plan

Si error rate > 5% o latency > +20%:

```
# Rollback inmediato
providers:
  mulesoft:
    enabled: false # Disable MuleSoft
    canary-percentage: 0
```

Tiempo de rollback: < 5 minutos

## Fase 4: Migración Completa & Cleanup

Status:  NOT STARTED

Duración Estimada: 1 semana

Esfuerzo: 20-30 horas

### Tasks

#### 1. Route 100% tráfico a MuleSoft (Day 1)

```
providers:
  mulesoft:
    enabled: true
    canary-percentage: 100
```

#### 2. Disable legacy providers (Day 2)

```
providers:
  twilio:
    enabled: false # DEPRECATED
  push:
    enabled: false # DEPRECATED
  voice:
    enabled: false # DEPRECATED
  biometric:
    enabled: false # DEPRECATED
```

### 3. Delete legacy adapter code (Day 3-4)

#### Archivos a eliminar:

- TwilioSmsProvider.java → DELETE (~300 LOC)
- TwilioVoiceProvider.java → DELETE (~350 LOC)
- FcmPushProvider.java → DELETE (~250 LOC)
- BiometricStubProvider.java → DELETE (~100 LOC)
- Tests asociados → DELETE (~600 LOC)

#### Dependencias a eliminar en pom.xml:

```
<!-- DELETE -->
<dependency>
  <groupId>com.twilio.sdk</groupId>
  <artifactId>twilio</artifactId>
</dependency>
<dependency>
  <groupId>com.google.firebase</groupId>
  <artifactId>firebase-admin</artifactId>
</dependency>
```

**Total LOC eliminado:** ~2,000 líneas

### 4. Update documentation (Day 5)

- Architecture diagrams
- README.md
- OpenAPI specification
- Runbooks
- ADR-003 (marcar como IMPLEMENTED)

### 5. Cleanup configuration

- Remove Twilio secrets from Vault
- Remove FCM service account JSON
- Archive legacy provider configs
- Update monitoring dashboards



Estimación de Esfuerzo

Desglose por Fase

Fase	Duración	Esfuerzo (horas)	Story Points	Costo Estimado*
Fase 0: MVP	10 semanas	<div><div></div>COMPLETADO</div>	N/A	<div><div></div>DONE</div>
Fase 1: API Design	2-3 semanas	40-60	N/A	€4,000-€6,000
Fase 2: Implementation	2 semanas	80-100	8 SP	€8,000-€10,000
Fase 3: Canary Deployment	2 semanas	60-80	5 SP	€6,000-€8,000
Fase 4: Cleanup	1 semana	20-30	2 SP	€2,000-€3,000
TOTAL	7-8 semanas	200-270 horas	15 SP	€20,000-€27,000

\*Costo estimado basado en €100/hora (rate promedio senior developer)

Recursos Necesarios

Rol	Dedicación	Fases
Senior Backend Developer	100%	Todas
MuleSoft Integration Specialist	50%	Fase 1, 2
DevOps Engineer	25%	Fase 3, 4
QA Engineer	50%	Fase 2, 3

Rol	Dedicación	Fases
Architect	10%	Todas (review)

## Beneficios de la Migración

### Beneficios de Negocio

Beneficio	Impacto	Cuantificación
Gestión centralizada	Alto	Un equipo gestiona todas las integraciones
Optimización de costos	Medio	MuleSoft negocia mejores rates con providers
Compliance & Governance	Alto	Audit logging centralizado, políticas estandarizadas
Faster time-to-market	Alto	Nuevos providers sin cambios en Signature Router

### Beneficios Técnicos

Beneficio	Impacto	Cuantificación
Desacoplamiento de SDKs	Alto	-2 dependencias (Twilio SDK, Firebase Admin SDK)
Codebase simplificado	Alto	-2,000 LOC (4 adapters eliminados)
Unified observability	Medio	Monitoreo centralizado en MuleSoft + Signature Router
API versioning	Medio	Cambios de provider APIs manejados por MuleSoft

### ROI Estimado

**Inversión:** €20,000–€27,000 (migración)

**Ahorro anual estimado:**

- Mantenimiento de SDKs: €5,000/año
- Costos de providers optimizados: €10,000/año
- Tiempo de desarrollo (nuevos providers): €8,000/año

Payback period: 10-13 meses

 **Riesgos y Mitigaciones**

Riesgo	Probabilidad	Impacto	Mitigación
MuleSoft Gateway downtime	Baja	Alto	Circuit Breaker en Signature Router, degraded mode
Latency incrementada	Media	Medio	Monitor P99, optimize MuleSoft routing, regional deployments
MuleSoft rate limiting	Baja	Medio	Align retry policies, monitor 429 responses
Pérdida de features específicas	Media	Bajo	Document feature parity con MuleSoft team
API contract breaking changes	Baja	Alto	Contract tests (Pact), API versioning
Retraso en MuleSoft specs	Alta	Medio	MVP ya operativo con providers directos

 **Criterios de Éxito**

**Fase 2 (Implementation)**

- ☐ MuleSoftApiProvider pasa 100% unit tests
- ☐ Coverage > 95%
- ☐ Integration tests pasan con WireMock
- ☐ Contract tests pasan con Pact
- ☐ Performance tests: P99 latency < 500ms

**Fase 3 (Canary)**

- ☐ Canary 10% → 0 errores en 48h
- ☐ Canary 100% → 0 errores en 1 semana
- ☐ Latency regression < 10%
- ☐ Error rate parity (MuleSoft ≤ directo)

☐ Zero customer complaints

Fase 4 (Cleanup)

- ☐ 100% tráfico en MuleSoft
- ☐ Legacy code eliminado
- ☐ Documentación actualizada
- ☐ Team entrenado en MuleSoft integration
- ☐ Runbooks validados

 Timeline Propuesto

Opción 1: Inicio Inmediato (si MuleSoft specs disponibles)

Milestone	Fecha	Responsable
Fase 1 Start: API Design	2025-12-02	MuleSoft Team
Fase 1 End: Contract ready	2025-12-20	MuleSoft + SR Team
Fase 2 Start: Implementation	2026-01-06	SR Team
Fase 2 End: Testing complete	2026-01-20	SR Team + QA
Fase 3 Start: Canary 10%	2026-01-21	DevOps + SR Team
Fase 3 Milestone: Canary 100%	2026-02-04	DevOps + SR Team
Fase 4: Cleanup complete	2026-02-11	SR Team

Go-Live Target: 11 de Febrero de 2026 (10 semanas desde hoy)

Opción 2: Esperar MuleSoft Specs (más realista)

Milestone	Fecha	Responsable
MuleSoft specs ready	2026-Q1 (TBD)	MuleSoft Team
Fase 1-4 execution	+8 semanas	SR Team
Go-Live	2026-Q2 (TBD)	All Teams

---

## Próximos Pasos Inmediatos

### Para MuleSoft Team

1. **Definir OpenAPI 3.0 specification** (Prioridad: Alta)
  - Endpoints para SMS, Voice, Push, Biometric
  - Authentication scheme (OAuth2 recommended)
  - Error codes estandarizados
  - SLAs (response times, availability)
2. **Proveer sandbox environment** (Prioridad: Alta)
  - Dev environment URL
  - Test credentials
  - API documentation
  - Postman collection

### Para Signature Router Team

1. **Preparar ambiente de desarrollo**
  - Setup VPN/network access a MuleSoft sandbox
  - Configure Vault secrets placeholders
  - Update CI/CD pipelines
2. **Iniciar contract tests** (antes de implementación)
  - Definir consumer contracts en Pact
  - Validar con MuleSoft team
  - Automatizar en CI/CD

---

## Conclusión

La migración a MuleSoft API Gateway es una **inversión estratégica** que:

- ✓ **Simplifica el codebase** (-2,000 LOC)
- ✓ **Desacopla de SDKs externos** (Twilio, Firebase)
- ✓ **Centraliza governance** (compliance, costs, monitoring)
- ✓ **Permite zero-downtime migration** (canary deployment)
- ✓ **Mantiene resilience patterns** (Circuit Breaker en SR)

**Recomendación:** Proceder con la migración cuando MuleSoft API specs estén disponibles. Mientras tanto, el MVP actual está **production-ready** y operativo.

**Preparado por:** Claude AI (Software Architect)  
**Fecha:** 28 de Noviembre de 2025  
**Versión:** 1.0  
**Próxima Revisión:** Cuando MuleSoft specs estén disponibles

 **Anexos**

**A. Checklist Pre-Migración**

- ☐ MuleSoft OpenAPI 3.0 spec disponible
- ☐ Sandbox environment provisto
- ☐ Authentication credentials generados
- ☐ Network access configurado (VPN)
- ☐ Contract tests definidos
- ☐ Grafana dashboards preparados
- ☐ Runbooks actualizados
- ☐ Team training completado

**B. Referencias**

- `docs/architecture/08-mulesoft-integration-strategy.md` - Estrategia detallada
- `docs/architecture/adr/ADR-003-mulesoft-integration.md` - Decisión arquitectónica
- `docs/architecture/09-test-strategy-mulesoft-migration.md` - Testing strategy

**C. Contactos**

Rol	Equipo	Email
MuleSoft Integration Lead	MuleSoft Team	<a href="mailto:mulesoft-team@company.com">mulesoft-team@company.com</a>
Signature Router Tech Lead	SR Team	<a href="mailto:sr-team@company.com">sr-team@company.com</a>
DevOps Lead	Platform Team	<a href="mailto:devops@company.com">devops@company.com</a>
Architect	Architecture Team	<a href="mailto:architects@company.com">architects@company.com</a>

